A BACTERIAL ALGORITHM FOR SURFACE MAPPING USING A MARKOV MODULATED MARKOV CHAIN MODEL OF BACTERIAL CHEMOTAXIS

Alaa A. Kharbouch, Maya R. Said, and Alan V. Oppenheim

Research Laboratory of Electronics
Massachusetts Institute of Technology
77 Massachusetts Ave, Cambridge, MA 02139
aak@mit.edu, mayasaid@mit.edu, avo@mit.edu

ABSTRACT

Bacterial chemotaxis refers to the locomotory response of bacteria to chemical stimuli, where the general biological function is to increase exposure to some substances while reducing exposure to others. In this paper, we introduce an algorithm for surface mapping based on a model of the biological signaling network responsible for bacterial chemotaxis. The algorithm tracks the motion of a bacteria-like software agent, referred to as a *bacterial agent*, on an objective function. Results from simulations using one- and two-dimensional test functions show that the surface mapping algorithm produces an informative estimate of the surface, revealing some of its key characteristics. We also present a modification of the algorithm in which the software agent is given the ability to reduce the value of the surface at locations it visits (analogous to a bacterium consuming a substance as it moves in its environment) and show that it is more effective in reducing the surface integral within a certain period of time than a bacterial agent lacking the ability to sense surface information or respond to it.

Index Terms— biological control systems, estimation, biological system modeling

1. INTRODUCTION

Many algorithms draw their inspiration from phenomena in nature. Examples include genetic algorithms, simulated annealing, and artificial neural networks. In a similar spirit, this paper explores the possibility of exploiting the ability of bacteria to seek out higher or lower concentrations of certain substances, referred to as *chemotaxis* [1], for the problem of surface mapping. In particular, we are interested in estimating a function where no direct access to any values of the function is possible, i.e. none of the function values are globally available. This is in contrast to many machine learning applications where an algorithm attempts to learn a function from examples, i.e. from knowledge of the values the function takes at a limited number of points in the function domain [2]. The surface mapping algorithm proposed in this paper tracks a software agent which simulates a model of the biochemical network that controls chemotaxis as it explores the domain of the function of interest and samples it locally. The algorithm estimates the unknown scalar-valued function by tracking the movement of the software agent. This is formulated by analogy to the way a bacterium, such as *E.coli*, travels in an environment where the concentration varies as a function of spatial coordinates.

Bacteria move in their environment in an informed way, seeking higher concentrations of attractants and lower concentrations of repellents [3]. Their movement is characterized by two modes of operation: runs and tumbles. Runs correspond to periods of smooth forward swimming and are induced by a counterclockwise rotation of the bacterium's helical tails (motors) referred to as *flagella*. In contrast, tumbles correspond to random re-orientations of the cell direction with little or no displacement and occur when the flagella turn clockwise [4]. *E.coli* bacteria have about 5 to 10 flagellar motors per cell [3]. The movement of an *E.coli* cell in its environment is characterized by runs interrupted by tumbles. At a high level, *E.coli* motion can be modeled as a biased random walk where the cell compares external environmental conditions at different time instances and adjusts its swimming behavior accordingly. If an increasing attractant concentration is sensed, tumbling is suppressed and the cell is more likely to move further in that favorable direction. If a decreasing attractant concentration is sensed, tumbling is favored and the cell tends to turn away from that direction.

The protein network underlying *E.coli* chemotaxis is one of the most well-studied signaling pathways in cell biology. A mechanistic model of the protein network underlying bacterial chemotaxis was recently proposed using a framework referred to as Markov modulated Markov chains [5]. This framework allows modeling at multiple resolutions by simultaneously studying the fluctuations of signaling pathways (stochastic behavior) and computing their average behavior. In the context of bacterial chemotaxis, it allows a more detailed simulation of bacterial motion capturing some of the richer behavior such as adaptation (the process by which the bacterial response adapts back to its pre-stimulus response after a sustained constant stimulus). We therefore use the Markov modulated Markov chains model as a basis for the bacterial algorithm presented here. Note that the main goal in this paper is to develop a surface mapping algorithm based on a chemotactic strategy, and that the biological model is used towards this end.

In the next section, we describe the surface mapping algorithm by specifying the software agent and describing how it moves to different locations on a surface. Simulation results using one- and two- dimensional surfaces are then presented in Section 3.

2. BACTERIAL ALGORITHM FOR SURFACE MAPPING

The bacterial algorithm for surface mapping (BASM) is based on simulations of a bacteria-like software agent which implements the *a priori* Markov modulated Markov chains (a3MC) model of the

This work was supported in part by the Texas Instruments Leadership University Program, BAE Systems PO 112991, and Lincoln Laboratory PO 3077828.

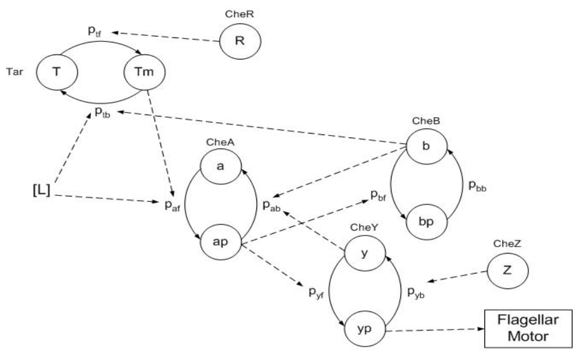


Fig. 1. Topology of the Markov modulated Markov chains model governing the surface mapping agent

E. Coli chemotaxis network presented in [5]. Figure 1 presents the topology of the underlying network where Tar, CheA, CheB, CheR, CheY, and CheZ refer to the proteins of interest. The flagellar motor (not shown in the figure) is represented by an eight-state Markov chain. The reader is referred to [5, 6] for details regarding the model. BASM simulates trajectories of this bacteria-like software agent in the domain of a surface represented by a function $C(\cdot)$. Although this paper will often discuss the mapping of a concentration function $C(\cdot)$, whose arguments are spatial coordinates, they are generally not restricted as such. The surface mapping agent, which we will refer to as a *bacterial agent*, switches between run and tumble modes to explore the surface using the surface values at different locations for guidance. The algorithm computes an estimate of the surface based on the movement of the bacterial agent. We next describe the algorithm steps. A more detailed description can be found in [6, 5].

2.1. Algorithm Steps

An important aspect of the exploration of the surface by the bacterial agent is the flagellar motor representation. With the input concentration to the a3MC model (denoted by $[L]$ in Figure 1) set as the surface value at the current location, the state probabilities for all the chains in the model are calculated. An a3MC stochastic simulation is performed with 9 flagellar motor chains, each of which can be in any of 8 states. The states are updated at every time step and the motors are simulated independently from each other. Using the voting hypothesis in [7, 4], the bacterial agent assumes a run or tumble state.

The motion and position of the bacterial agent in a two-dimensional search space at a time *n* is described in terms of its current position coordinates $x[n]$ and $y[n]$, its angular direction $\theta[n]$, and the state of the motors (run or tumble state). For each tumble, the bacterial agent rotates at a constant radial speed rs (in rad/s) counterclockwise or clockwise with equal probability. Thus, for every time step in tumble mode, $\theta[n]$ is adjusted as follows:

$$\theta[n+1] = \theta[n] + \mathrm{sgn}(U) \times rs \times dt \qquad (1)$$

where *dt* is the sampling time for the discrete-time Markov chains, set to $10^{-3}s$ in our simulations, and U is a random variable uniformly distributed over the interval [-1,1]. The position updates for a run are given by:

$$x[n+1] = x[n] + \cos(\theta[n]) \times v \times dt \qquad (2)$$

$$y[n+1] = y[n] + \sin(\theta[n]) \times v \times dt \qquad (3)$$

where *v* is the running velocity in (length unit)/s.

In the one-dimensional case, $\theta[n]$ determines whether the displacement is in the positive or negative *x* direction as follows:

$$x[n+1] = x[n] + \mathrm{sgn}(\cos(\theta[n])) \times v \times dt \qquad (4)$$

After initializing all variables at the beginning of a simulation, the algorithm steps can be summarized as follows:

1: Determine ($x[n]$,$y[n]$ and $\theta[n]$) using ($x[n-1]$,$y[n-1]$ and $\theta[n-1]$) and previous overall motor state according to equations (1)-(4).

2: Determine the current ligand input concentration for the agent as $g[n] = C(x[n])$ or $g[n] = C(x[n], y[n])$ in the case of one- or two-dimensional surfaces respectively.

3: Use concentration input and internal state values at previous time step $n-1$ to calculate current internal state probabilities (for time index n).

4: Use previous internal states to check for transitions in any of the 9 motors and determine new motor state.

5: Go to 1 for next time step.

2.2. Calculation of the Density Function

The algorithm produces a histogram indicating the relative amount of time the bacterial agent spent at every position. We refer to this as the density function. An estimate of the function is obtained by tracking the motion of multiple bacterial agents. Density functions from multiple simulations are averaged to calculate the estimate. Since the bacterial agents tend to seek out higher values of the concentration function, areas with higher densities indicate an increased likelihood of higher values near these locations.

For one-dimensional surface mapping simulations a uniform grid of the spatial dimension is formed, where each point on the grid is defined as:

$$x_k = k \times v \times dt \qquad (5)$$

for integer k, such that the points are separated by $\triangle_x = vdt$. Every simulation produces a density function $D[k]$ corresponding to the total number of time steps during the entire simulated movement trajectory where the bacterial agent was in the run state, and in a position that falls within the bin (of size $\triangle_x$) centered at x_k.

For two-dimensional surface mapping simulations a two-dimensional uniform grid of the spatial dimensions is formed, with each point on the grid written as (x_{k_1},y_{k_2}) where $x_{k_1} = k_1\triangle_x$ and $y_{k_2} = k_2\triangle_y$ such that adjacent points on the grid are separated by $\triangle_x$ and $\triangle_y$ distance units in the horizontal and vertical dimensions respectively. Every simulation produces a density function $D[k_1, k_2]$.

3. SIMULATIONS AND RESULTS

The surface mapping algorithm was evaluated by running simulations on one- and two-dimensional test functions. The average density function was computed for different values of parameters v and rs and was compared to the test function.

3.1. One-Dimensional Surface Mapping Simulations

One-dimensional BASM simulations were performed using a unimodal test surface $C_1(x) = 10^{-4}\exp(-\frac{3}{4}|x|)$ and a multimodal test surface $C_2(x) = (5 \times 10^{-5})\left|\frac{\sin(\frac{1}{4}x)}{\frac{1}{4}x}\right|$. 40 simulations, 1000 seconds each, were run. The bacterial agent position at the start of the simulation was initialized randomly according to a uniform distribution between -10 and 10. The initial direction, $\theta[0]$, was set to 0 or π with equal probability. For BASM simulations on the unimodal concentration surface $C_1(x)$, a default run speed *v* of 0.75 s^{-1} and a tumbling rotation speed *rs* of π rad/*s* were used. For the multimodal function $C_2(x)$, the run speed and tumbling rotation were $v = 0.6s^{-1}$ and $rs = \pi$ rad/*s* respectively. The average density function was then smoothed using a 2001-point FIR averaging filter and normalized such that the integral is one. The results are shown in figures 2(a) and 2(d). The density function provides an approximation of the shape of the test function including the location of its maxima and minima. Both density functions are approximately symmetric, and the density function from the simulations using the multimodal surface captures the main lobe and the adjacent side lobes of lower height. The global maximum of the smoothed density function for the unimodal surface simulations was found to be at $x = -0.228$, less than a quarter of a distance unit away from the true maximum of the surface.

Figures 2(b) and 2(e) show the results obtained when the running speed *v* is increased for the $C_1(x)$ and $C_2(x)$ simulations to $1.25s^{-1}$ and $1s^{-1}$ respectively. We can see that the density function is more spread out, as the higher run speed allows the bacterial agent to spend more time exploring areas further away from both its starting position and the global maximum at zero. We also observe that the mapping of $C_2(x)$ captures the main lobe of the sinc function as well as three of the side lobes on either side of it. Figures 2(c) and 2(f) show the results obtained when the rotation speed *rs* is increased for the $C_1(x)$ and $C_2(x)$ simulations to $\frac{3\pi}{2}$ rad/*s* and $\frac{5\pi}{4}$ rad/*s* respectively. This has the opposite effect on the density function than increasing *v*, as the density suggests that the bacterial agent did not spend a significant amount of time outside a small range around the global maximum. This is expected as the high tumbling rotation speed results in more frequent switches in running direction and the bacterial agent is therefore not likely to travel far in one direction.

3.2. Two-Dimensional Surface Mapping Simulations

Two-dimensional BASM simulations were also performed using two test concentration surfaces that are two-dimensional extensions of $C_1(x)$ and $C_2(x)$:

$$C_3(x,y) = 10^{-4}\exp(-\frac{1}{2}\sqrt{x^2+y^2}) = (10^{-\frac{4}{3}})(C_1(\sqrt{x^2+y^2}))^{\frac{2}{3}} \tag{6}$$

$$C_4(x,y) = (5\times10^{-5})\left|\frac{\sin(\frac{1}{4}x)\sin(\frac{1}{4}y)}{\frac{1}{16}xy}\right| = (5\times10^{5})C_2(x)C_2(y) \tag{7}$$

The multimodal function $C_4(x,y)$ is shown in Figure 3(a). 80 simulations, 2000 seconds each, were performed with spatial spacing parameters $\triangle_x$ and $\triangle_y$ set to 2. The initial position of the bacterial agent was set to be a distance R away from the global maximum at the origin, i.e. $x[0] = R\cos(\Theta_i)$ and $y[0] = R\sin(\Theta_i)$ where Θ_i is a random number uniformly distributed between 0 and 2π. The initial angular direction $\theta[0]$ was also set to an independent random number Φ_i with the same distribution. The results presented in this section are from simulations that use an R of 7. A default run speed *v* of 0.3 s^{-1} and a tumbling rotation speed *rs* of 10π rad/*s* were used. The average density functions obtained are shown in Figure 3(b) for $C_3(x,y)$ and 3(c) for $C_4(x,y)$. In the unimodal case, there is a single clear peak in the density function. In the multimodal mapping, we can see the main lobe of the two-dimensional sinc function, as well as the four neighboring side lobes. The oscillating nature of the function is conveyed through the density function, and it is evident that the majority of the highest peaks lie along the x- and y-axes.

3.3. Surface Flattening

In this section, we explore the use of bacterial agents that actively modify the function landscape as they use the information they gather about the surface to bias their exploration. Specifically, bacterial agents are allowed to reduce the value of a function at the locations they visit. They essentially flatten the concentration surface to approximately zero as they visit areas with higher concentrations more often and reduce the total amount of the substance.

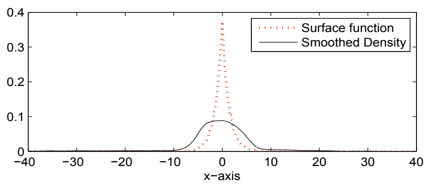
In the one-dimensional surface flattening algorithm, a uniform grid of the x-axis is formed as in equation (5) for $|x| \leq x_{max}$ or equivalently, $|k| < \frac{x_{max}}{vdt}$. The time-varying concentration function is denoted by $C[k,n]$, where *k* is a spatial index and *n* is the time index. At the beginning of the simulation $C[k,n]$ is initialized to a sampled version of the concentration surface $C(x)$, i.e. $C[k,0] = C(x_k)$. For every time step, if the flagellar motors are in a run mode, the current position $x[n]$ is rounded to the nearest value of x_k and the discrete-space concentration function $C[k,n]$ is reduced at the corresponding k by a factor of γ:

$$C[k,n+1] = \gamma C[k,n] \tag{8}$$

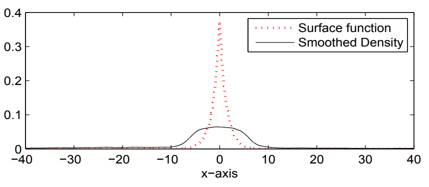
The surface value the bacterial agent reads at that location is the value of the time-varying surface (interpreted as a concentration) at the nearest grid point, i.e. $C[k,n]$. Any position updates due to a run that would lead to a value of $x[n]$ outside the range $|x| < x_{max}$ are prevented. We set x_{max} to 90 for the one-dimensional simulations. We investigate whether the bacterial agent is more effective at reducing the total amount of a substance than an unbiased version of the random walk, implemented using a bacterial agent that always senses a complete lack of attractant everywhere (zero concentration), and is therefore not influenced by the concentration surface. Two sets of simulations are performed. For each simulation, the total amount of remaining substance, denoted by $S[n]$, is used as a metric of how fast the two algorithms flatten the surface. The surface sums from 20 simulations are averaged to obtain an estimate of the expected amount of remaining substance at time n as follows:

$$S_{avg}[n] = \frac{1}{20}\sum_{i=1}^{20} S_i[n] = \frac{1}{20}\sum_{i=1}^{20}\sum_{k=-\frac{x_{max}}{vdt}}^{\frac{x_{max}}{vdt}} C_i[k,n] \tag{9}$$

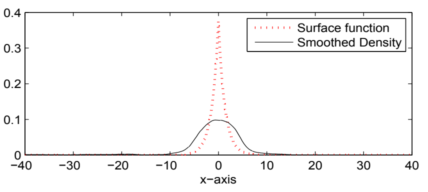
where i denotes the simulation number. We use $C_2(x) = (5\times10^{-5})\left|\frac{\sin(\frac{1}{4}x)}{\frac{1}{4}x}\right|$ as the surface, a *v* of 0.75 s^{-1}, a tumbling rotation speed *rs* of π rad/*s*, and a reduction factor γ of 0.8. Figure 4(a) shows the calculated average running sum of the surface for the surface biased and unbiased versions of the random walk. The surface flattening algorithm can be extended to two-dimensional surfaces. In this case, the time-varying surface is a function of two spatial indices k_1 and k_2, in the two-dimensional simulations they are related to the continuous spatial variables according to $x_{k_1} = k_1 \times 50 \times v \times dt$ and $y_{k_2} = k_2 \times 50 \times v \times dt$. Any position updates due to a run that would lead to a value of $x[n]$ or $y[n]$ outside the range $|x| \leq x_{max}$, $|y| \leq y_{max}$ are prevented. x_{max} and y_{max} are set to 90 for the two-dimensional simulations. The surface sums from 20 simulations are averaged to calculate $S_{avg}[n]$. We use $C_4(x,y) = (5\times10^{-5})\left|\frac{\sin(\frac{1}{4}x)\sin(\frac{1}{4}y)}{\frac{1}{16}xy}\right|$ as the surface, and set $v = 5s^{-1}$, $rs = 5\pi$ rad/*s*, and $\gamma = 0.8$. Figure 4(b) shows the calculated average running sum of the surface for the surface biased and unbiased versions of the random walk.



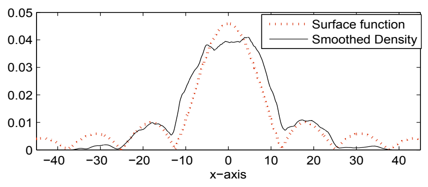
(a) $C(x)$=$C_1(x)$, v=$0.75s^{-1}$, rs=π



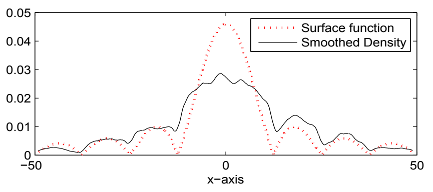
(b) $C(x)$=$C_1(x)$, v=$1.25s^{-1}$,rs=π



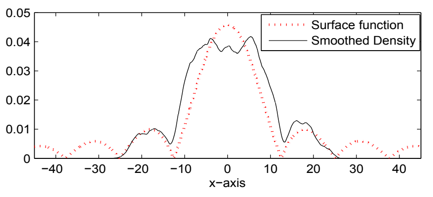
(c) $C(x)$=$C_1(x)$, v=$0.75s^{-1}$, rs=$\frac{3}{2}\pi$



(d) $C(x)$=$C_2(x)$, v=$0.6s^{-1}$, rs=π

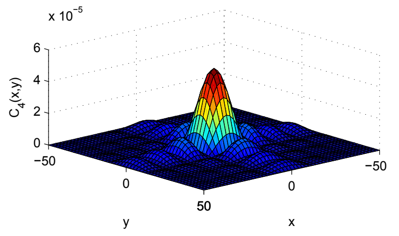


(e) $C(x)$=$C_2(x)$, v=$1s^{-1}$,rs=π

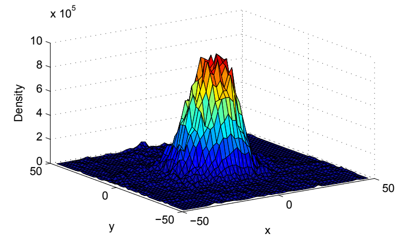


(f) $C(x)$=$C_2(x)$, v=$0.6s^{-1}$, rs=$\frac{5}{4}\pi$

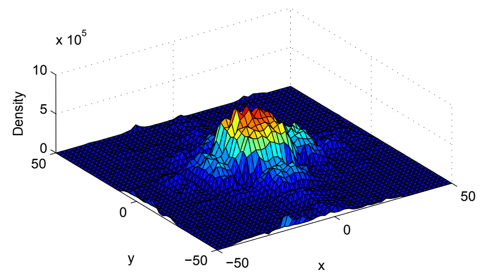
Fig. 2. Smoothed density function from surface mapping simulations using different values of v and rs



(a) $C_4(x, y)$



(b) Average density $C(x, y)$=$C_3(x, y)$



(c) Average density $C(x, y)$=$C_4(x, y)$

Fig. 3. The two-dimensional test surface $C_4(x, y)$, and average density function results from two-dimensional surface mapping simulations

4. REFERENCES

[1] S. Mueller, J. Marchetto, S. Airaghi, and P. Koumoutsakos, "Optimization based on bacterial chemotaxis," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 1, 2002.

[2] P. Magni, R. Bellazzi, and G. De Nicolao, "Bayesian function learning using mcmc methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, 1998.

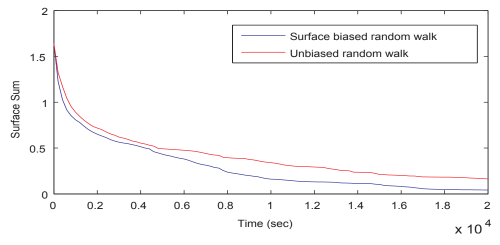
[3] A. Bren and M. Eisenbach, "How signals are heard during bacterial chemotaxis: protein-protein interactions in sensory signal propagation.," *J. Bacteriol.*, vol. 182, no. 24, 2000.

[4] P.A. Spiro, J.S. Parkinson, and H.G. Othmer, "A model of excitation and adaptation in bacterial chemotaxis," *Proc. Natl. Acad. Sci. USA*, vol. 94, no. 14, 1997.

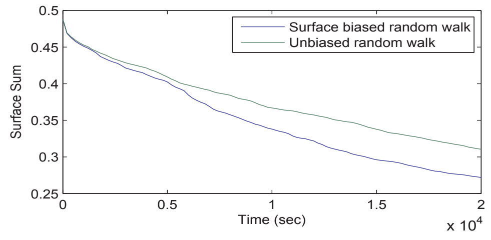
[5] M.R. Said, *Signal Processing in Biological Cells: Proteins, Networks, and Models.*, Ph.D. thesis, MIT, 2005, Also available as RLE technical report No. 711, June. 2006, MIT.

[6] A.A. Kharbouch, *A Bacterial Algorithm for Surface Mapping Based on a Markov Modulated Markov Model of Chemotaxis.*, M.S. thesis, MIT, 2006.

[7] A. Ishihara, J.E. Segall, S.M. Block, and H.C. Berg, "Coordination of flagella on filamentous cells of *escherichia coli*," *J. Bacteriol.*, vol. 155, no. 1, 1983.



(a) One-dimensional surface sum



(b) Two-dimensional surface sum

Fig. 4. Average surface sum results from surface flattening simulations using the surface biased and unbiased versions of the random walk.